

Architecture MVC

May 14, 2024

Application classique

Deux grandes parties :

- ▶ La partie visible (front office) : l'IHM (Interface Homme-Machine).
 1. Les entrées : actions de l'utilisateur.
 2. Les sorties : la visualisation des résultats des actions de l'utilisateur.
- ▶ La partie cachée (back office) : Le stockage, l'accès et le traitement des données : Coeur de l'application.

Architecture MVC

- ▶ M : Modèle (Model).
- ▶ V : Vue (View).
- ▶ C : Contrôleur (Controller).

Architecture MVC : Généralités

- ▶ Proposée en 1979 par Trygve M. H. Reenskaug.
- ▶ Fortement liée à la programmation objet et maintenant aux applications Web (Symfony, Laravel, etc).
- ▶ Patron de conception (design pattern), c'est-à-dire une solution standard pour répondre à un problème.
- ▶ Architecture logicielle pour structurer une application.

Architecture MVC : Pourquoi ?

- ▶ Problème :
 - ▶ Comment rendre le modèle (domaine métier) indépendant des vues (interface utilisateur) qui en dépendent ?
 - ▶ Réduire le couplage entre modèle et vue
- ▶ Solution :
 - ▶ Insérer une couche supplémentaire (contrôleur) pour la gestion des évènements et la synchronisation entre modèle et vue.

Modèle

Le Modèle :

- ▶ Implémente le fonctionnement du système.
- ▶ Gère les accès aux données métiers.

C'est à dire :

- ▶ Représentation des données.
- ▶ Accès aux données.
- ▶ Traitement des données.

La Vue :

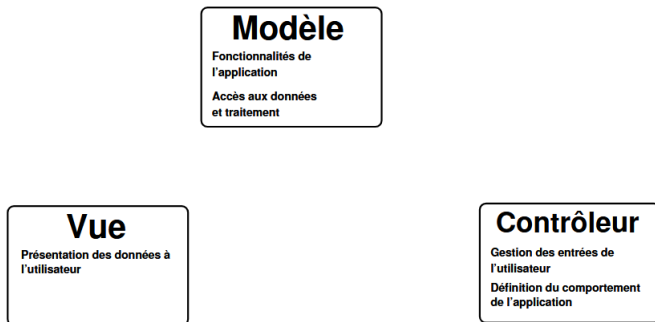
- ▶ Présente les données en cohérence avec l'état du modèle.
- ▶ Capture et transmet les actions de l'utilisateur.

Contrôleur

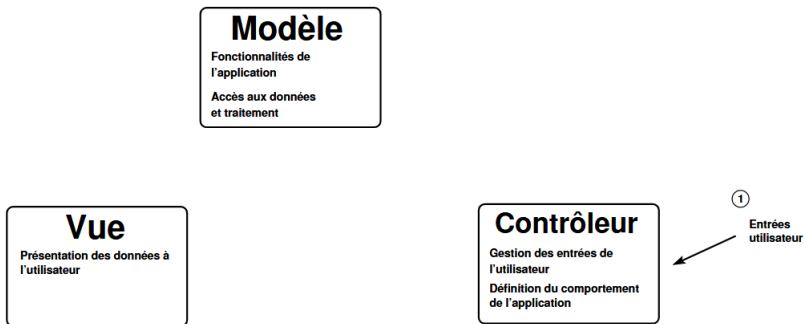
Le Contrôleur :

- ▶ Gère les changements d'état du modèle.
- ▶ Informe le modèle des actions utilisateur.
- ▶ Sélectionne la vue appropriée (programmation événementielle).

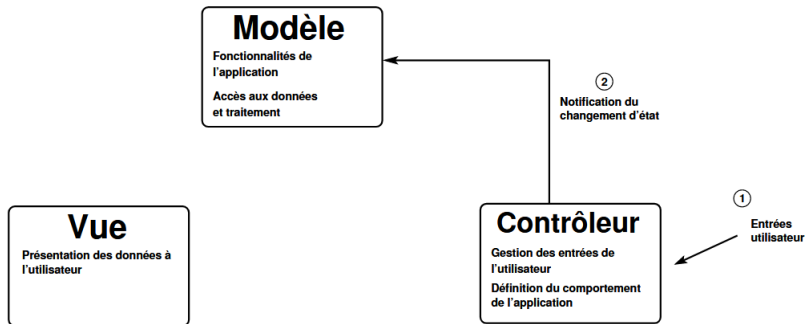
Fonctionnement



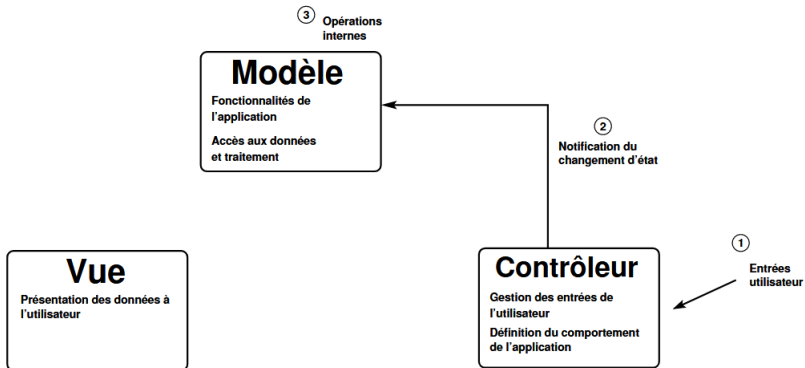
Fonctionnement



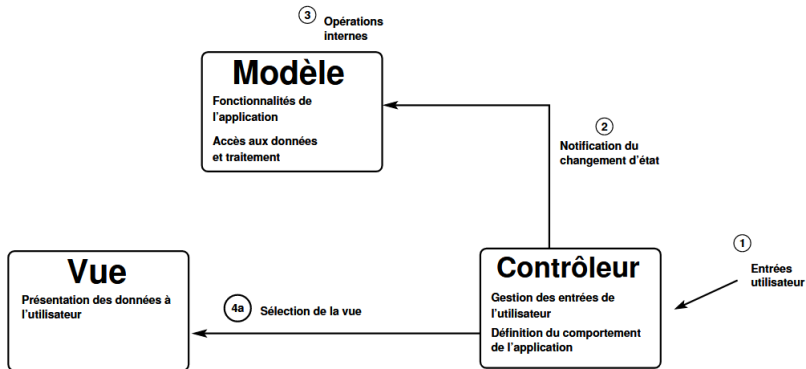
Fonctionnement



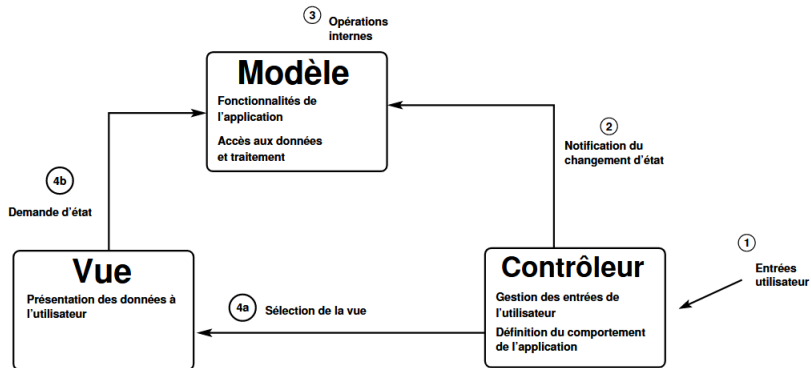
Fonctionnement



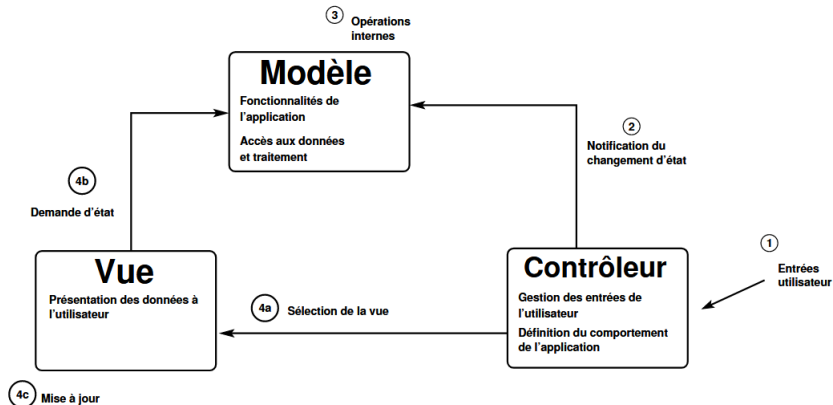
Fonctionnement



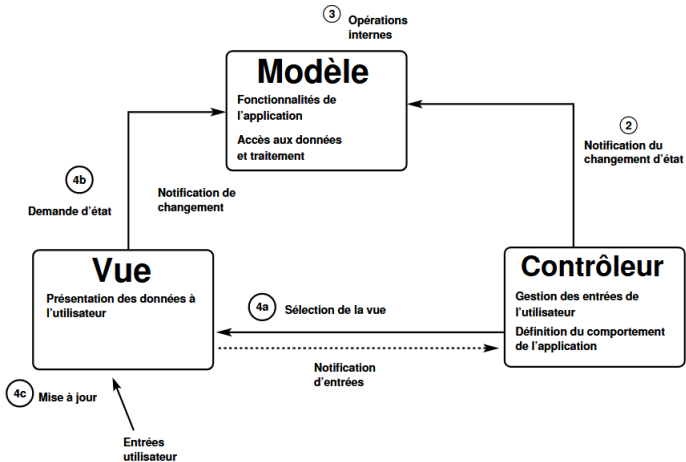
Fonctionnement



Fonctionnement



Fonctionnement



Avantages

- ▶ Structuration claire de l'application.
- ▶ Indépendance entre les données, la visualisation et le traitement.
- ▶ Réutilisation possible des éléments.

Inconvénients

- ▶ Complexité accrue.
- ▶ Augmentation du temps d'exécution.

Autres alternatives

Architecture en couche

- ▶ C'est le cas du réseau (Modèle OSI et TCP/IP)
- ▶ Chaque couche interagit avec ses voisins. Exemple :